A Bayesian 3D Search Engine using Adaptive Views Clustering

Tarik Filali Ansary, Mohamed Daoudi, Jean-Phillipe Vandeborre

FOX-MIIRE Research Group (LIFL UMR USTL-CNRS 8022)

(GET/INT - ENIC Telecom Lille 1)

email: {filali, daoudi, vandeborre}@enic.fr

http://www-rech.enic.fr/miire

Abstract

In this paper, we propose a method for 3D model indexing based on 2D views, named AVC (Adaptive Views Clustering). The goal of this method is to provide an "optimal" selection of 2D views from a 3D model, and a probabilistic Bayesian method for 3D model retrieval from these views. The characteristic views selection algorithm is based on an adaptive clustering algorithm and use statistical model distribution scores to select the optimal number of views. Starting from the fact that all views do not have equal importance, we also introduce a novel Bayesian approach to improve the retrieval. We finally present our results and compare our method to some state of the art 3D retrieval descriptors on the *Princeton 3D Shape Benchmark* database and a 3D CAD models database supplied by the car manufacturer *Renault*.

1 Introduction

The use of three-dimensional image and model databases throughout the Internet is growing both in number and size. The development of modeling tools, 3D scanners, 3D graphic accelerated hardware, Web3D and so on, is enabling access to three-dimensional materials of high quality. In recent years, many systems have been proposed for efficient information retrieval from digital collections of images and videos. However, the solutions proposed so far to support retrieval of such data are not always effective in application contexts where the information is intrinsically three-dimensional. A similarity metric has to be defined to compute a visual similarity between two 3D models, given their descriptions.

For example, Kazhdan et al. [1] describe a general approach based on spherical harmonics. From the collection of spherical functions calculated on the voxel grid of the 3D object, they compute a rotation invariant descriptor by decomposing the function into its spherical harmonics and summing the harmonics within each frequency. Then, they compute the L_2-norm for each component. The result is a 2D histogram indexed by radius and frequency.

Vandeborre et al. [2] propose to use full three-dimensional information. The 3D objects are represented as mesh surfaces and 3D shape descriptors are used. The results obtained show the limitation of the approach when the mesh is not regular. This kind of approach is not robust in terms of shape representation.

Antini et al [3] present an approach based on curvature correlograms. The main advantage of correlograms relates to their ability to encode not only the distribution of features but also their arrangement on the object surface.

Assfalg et al. [4] present an approach to global and local content-based retrieval of 3D models that is based on Spin images. Spin images are used to derive a view-independent description of both database and query objects. A set of Spin images is first created for each object and the parts it is composed of. Then, a descriptor is evaluated for each Spin image in the set. Clustering is performed on the set of image-based descriptors of each object to achieve a compact representation.

Sundar et al. [5] intend to encode a 3D object in the form of a skeletal graph. They use graph matching techniques to match the skeletons and, consequently, to compare the 3D objects. They also suggest that this skeletal matching approach has the ability to achieve part matching and helps in defining the queries instinctively.

In 3D retrieval using 2D views, the main idea is that two 3D models are similar, if they look similar from all viewing angles. Funkhouser et al. [6] apply view based similarity to implement a 2D sketch query interface. In the preprocessing stage, a descriptor of 3D model is obtained by 13 thumbnail images of boundary contour as seen from 13 view directions.

Filali et al. [7] [8] propose an adaptive nearest neighbor like framework to choose the characteristic views of a 3D model. The framework gives good results but was experimented on a small database.

Using aspect graphs, Cyr and Kimia [9] specify a query by a view of 3D objects. A descriptor of 3D model consists in a set of views of the 3D models. The number of views is kept small by clustering views and by representing each cluster with one view, which is represented by a shock graph. Schiffenbauer [10] presents a complete survey of aspect graphs methods. Using shock matching, Macrine et al. [11] apply indexing using topological signatures vectors to implement view based similarity matching more efficiently.

Chen et al. [12] defend the intuitive idea that two 3D models are similar if they also look similar from different angles. Therefore they use 100 orthogonal projections of an object and encode them by Zernike moments and Fourier descriptors. They also point out that they obtain better results than other well-known descriptors.

At last, for a further read on 3D retrieval state of the art, Tangelder and Veltkamp [13] present a complete survey on 3D shape retrieval.

In this paper, we propose a method for 3D model indexing based on 2D views, named AVC (Adaptive Views Clustering). This method aims at providing an optimal selection of 2D views from a 3D model, and a probabilistic Bayesian method for 3D models indexing from these views. This paper is organized in the following way. In section 2, we present the main principles of our method for characteristic views selection. In section 3, our probabilistic 3D models indexing is presented. Finally, the results obtained from two databases of 3D models are presented showing the performances of our method. We compare our method to some state of the art 3D retrieval descriptors on the *Princeton 3D Shape Benchmark* database and the SEMANTIC-3D database.

2 Selection of characteristic views

Let $D_b = \{M_1, M_2, \ldots, M_N\}$ be a collection of N three-dimensional models. We wish to represent each 3D model M_i by a set of 2D views that best represent it. Figure 2 shows an overview of the selection of characteristic views algorithm. To achieve this goal, we first generate an initial set of views from the 3D model, then we reduce it to the only views that best characterize the 3D model. This idea comes from the fact that all the views of 3D model do not have equal importance: there are views that contain more information than others. For example, one view is sufficient to represent a sphere as it look the same from all the angles. But, more then one view is need to represent a more complex 3D model as an airplane.

In this paragraph, we present our algorithm for characteristic views selection from a three-dimensional model.

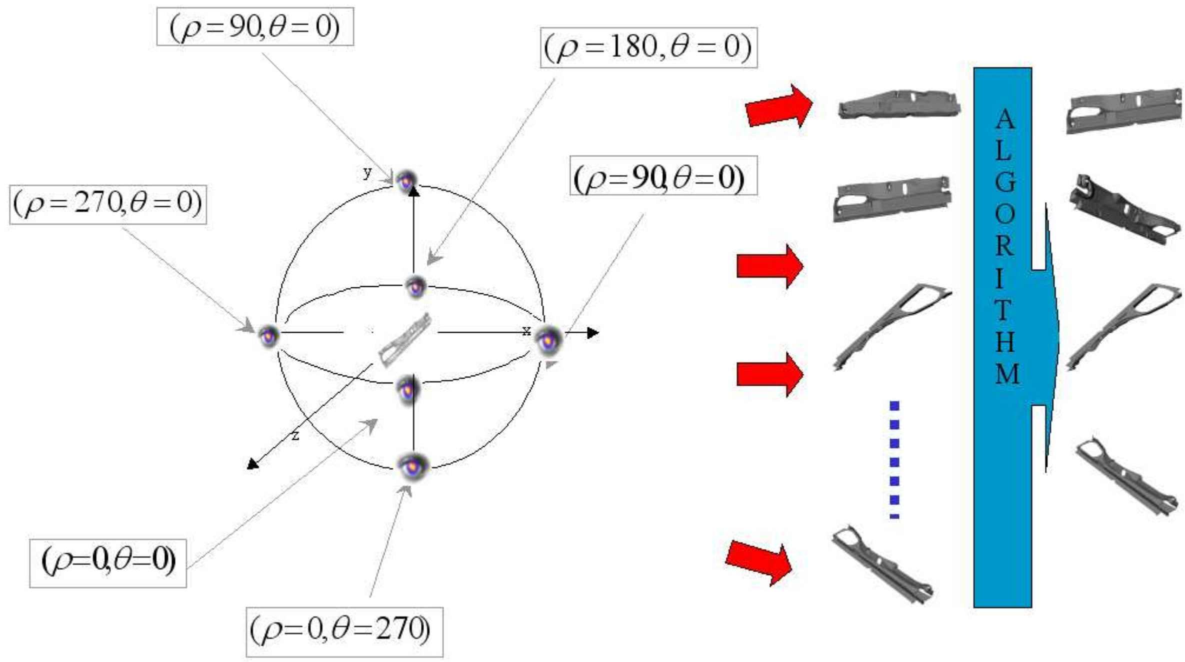


Figure 1: The views selection process

2.1 Generating the initial set of views

To generate the initial set of views for a model M_i of the collection, we create 2D views (projections) from multiple viewpoints. These viewpoints are equally spaced on the unit sphere. In our current implementation, we use 320 initial views. In fact we scale each of our model that it can fit in a unit sphere then we translate it, that the origin coincide with the 3D model barycenter.

To select positions for the views, that must be equally spaced, we use a two unit icosahedron centred on the origin. We subdivide the icosahedron once by using the Loop subdivision schema to obtain a 320 faceted polyhedron. Finally to generate the initial views we place the camera on each of the face-center of the polyhedron looking at the coordinate origin.

The views are silhouettes only, which enhance the efficiency and the robustness of image metric. Orthogonal projection is applied in order to speed up the retrieval process and reduce the size of the used features. To represent each of these 2D views, we use 49 coefficients of Zernike moment descriptor [14] [15]. Consequently to the use of Zernike moments, the approach is robust against translation, rotation and scaling.

2.2 Characteristic views selection

As every 2D view is represented by 49 Zernike moment coefficients, choosing a set of characteristic views that best characterize the 3D models (320 views), is equivalent to choose a subset of points that represent a set of 320 points in a 49 dimensions space. The problem of choosing X characteristic views that best represent a set of $N = 320$ views, is well known as *clustering problem.*

Data clustering is a well known problem in the mathematical and computer-science communities. The literature in this domain is huge. One of the widely used method is *K-means* [16]. Its attractiveness lies in its simplicity and in its local-minimum convergence properties. However, it has one main shortcomming: the number of clusters K has to be supplied by the user.

As we want from our method to adapt the number of characteristic views to the geometrical complexity of the 3D model, using K-means is not suited. To avoid this problem, we use a method derivative from K-means. Instead of a fixed number of clusters, we propose to use a range in which we will choose the best number of clusters. In our case, the range will be $[1, \ldots, 40]$. In this paper, we assume that the maximum number of characteristic views is 40. This number of views is a good compromise between speed, descriptor size and representation.

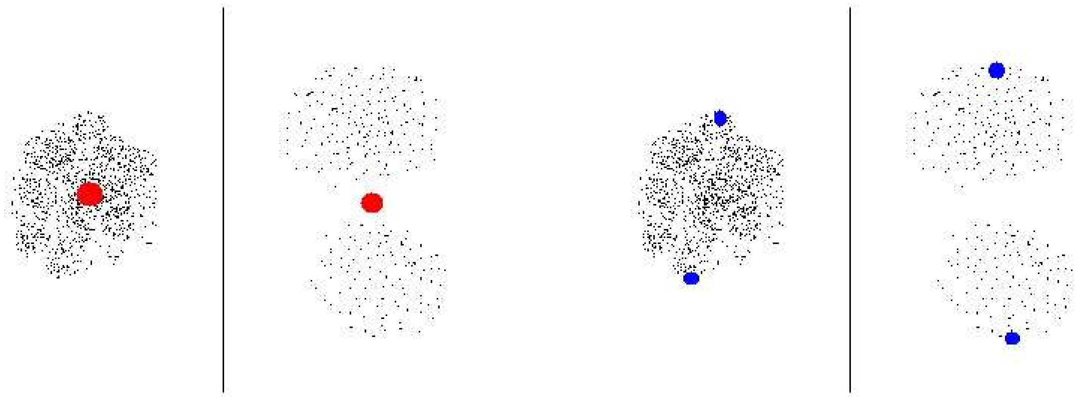
We proceed now to demonstrate how to select the characteristic views set and also how to select the best K within the given range. In essence, the algorithm starts with one characteristic view (K equal to 1), we add characteristic views where they are needed and we do a global K-means on the data starting with characteristic views as cluster centers. We continue alternating between adding characteristic views and doing a global K-means until the upper bound for characteristic views number (40) is reached. During this process, for each K, we save the characteristic views set.

To add new characteristic views, we use the idea presented in X-means clustering method by Dan Pelleg [17]. In a first step, in every cluster of views represented by a characteristic view, we select two views that have the maximum distance in this cluster. Next, in each cluster of views, we run a local K-means (with $K = 2$) for each pair of selected views. By local we mean that only the views that are in the cluster are used in this local clustering (Figure 2).

Please note that figure 2 and 3 are just a schematic example as we represent a view in a two dimensional space. In our system each view is represented by a vector in a 49 dimensional space (corresponding to the 49 Zernike moment features extracted from the view).

At this point, a question arises: "are the two new characteristic views giving more information on the region than the original characteristic view?". To answer this question, we use Bayesian Information Criteria (BIC) (section 4), that scores how likely the representation model (using one or two characteristic views) is fitting the data. Other criteria like Akaike Information Criteria (AIC) [18] could also be used. Estimating the BIC score will be discussed in the next section.

According to the outcome of the test, the model with the higher score is selected (Figure 3). These clusters of the views which are not represented well by the current centroids will receive more attention by increasing the number of centroids in them.



(a) (b)

Figure 2: Local k-means on each part of the views clusters with k=2

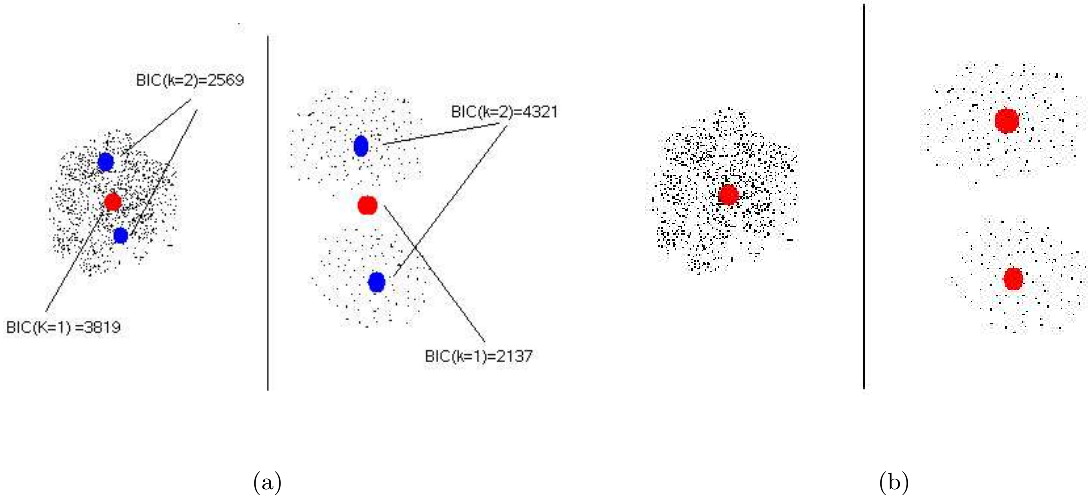


Figure 3: Selecting the representations (with 1 or 2 characteristic views) that have the higher bic score

We continue alternating between global K-means and K-means on clusters owned by characteristic views until the upper bound for characteristic views number is reached. Then we compare the BIC score of each characteristic views set. Finally, the best characteristic views set will be the one that gets the highest BIC score on all the views. Algorithm 1 gives an overview of the characteristic views selection algorithm.

Algorithm 1 characteristic views selection algorithm.

Number of characteristic views = 1

while Number of characteristic views < Maximum number characteristic views **do**

  Make global K-means on all the views (The start centers are the characteristic views).

  Save the characteristic views set and it's BIC Score.

  for all cluster of views **do**

    Make K-means (with K=2) on the cluster.

    Choose the representation with the higher BIC score. The original characteristic view or the two new characteristic views

    Update the number of characteristic views.

  end for

end while

Select the K and the characteristic view set with the higher BIC score.

2.3 Bayesian Information Criteria.

"Representation model selection" refers to the problem of using the data to select one model from a list of candidates $Mod_1, \ldots, Mod_k$ that represent best the data. By a "model" we mean a set of probability distributions. In our case, a representation model will be a set of characteristic views.

Model selection is well a known mathematical problem. This problem can be solved using many criteria like Akaike Information Criteria (AIC)[18], Bayesian Information Criteria(BIC)[19], Bayes Factors[20] and so on. The Bayesian solution to theses problems is to compute the posterior probability $P[Mod_j|D]$ for each model. Under weak conditions, BIC and Bayes methods are asymptotically equivalent. In the other hand it appears to be some debate on the relative merits of AIC versus BIC, but it's far behind the scope of this paper[21][22]. In practice, it has been observed that BIC select models with a dimension smaller than AIC, which is normal because BIC "penalize" more than AIC when dimension is higher then seven [23]. In this paper, we use Bayesian Information Criteria (BIC) for model selection.

To calculate the BIC score for a *representation model* Mod_j having the cluster of views V, we use the formula introduced by Schwarz [19]:

$$BIC(Mod_j) = \hat{l}_j(V) - \frac{P_j}{2} \; log \; N \; . \tag{1}$$

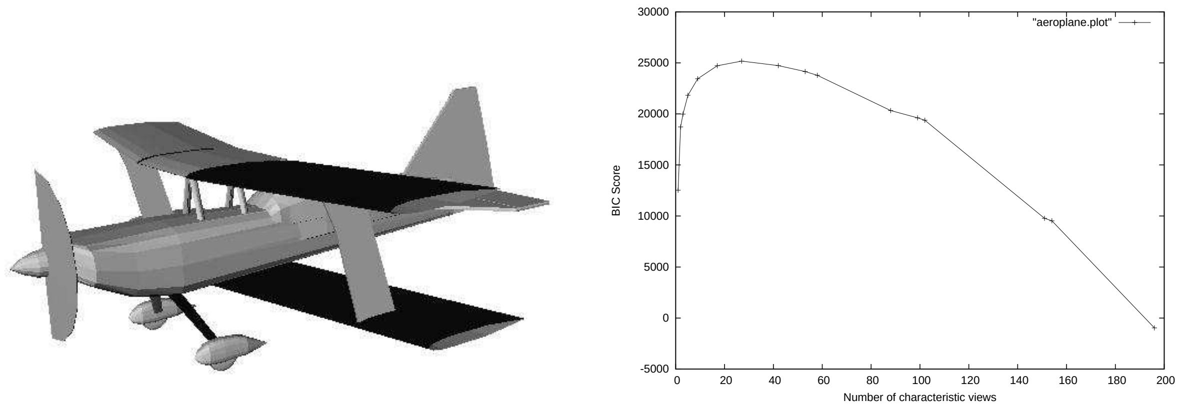
With P_j the number of parameters in Mod_j. This is also known as the Schwarz criterion [19]. $\hat{l}_j(V)$ is the log-likehood of the data according to the j-th model and taken at the maximum likelihood point. N is the number of views in the cluster $N = |V|$. In our case the models are all spherical Gaussians which is the type assumed by K-means. The Maximum Likehood Estimate (MLE) for variance is:

$$\hat{\theta}^2 = \frac{1}{N-K} \sum\nolimits_i (Dist(V_i, Vc_i)^2) \; . \tag{2}$$

With $Dist(V_i, Vc_i)$ the Euclidean distance between the Zernike moments of the respective views V_i and Vc_i. Where Vc_i is the characteristic view associated with the view V_i. The log-likehood of the data is:

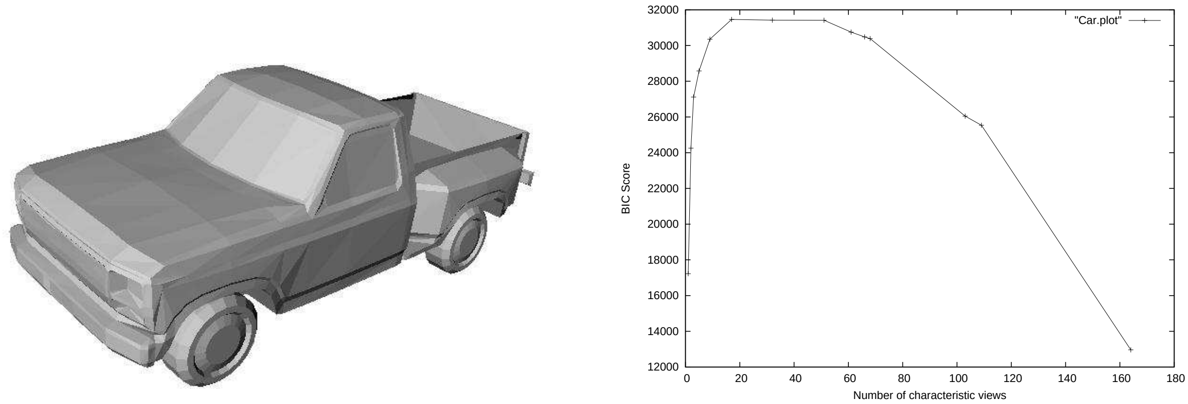
$$\hat{l}_j(V) = \sum\nolimits_i \left(\frac{1}{\sqrt{2\pi}\hat{\theta}^{49}} - \frac{1}{2\hat{\theta}^2} \parallel Dist(V_i, Vc_i) \|^2 + log \frac{N_{(i)}}{N} \right) \; . \tag{3}$$

Figure 4 and 5 show the evolution of the BIC score with the number of views. Theses curves



(a) (b)

Figure 4: Airplane model from the *Princeton 3D Shape Benchmark* and its corresponding BIC Score curve.



(a) (b)

Figure 5: A car model from the*Princeton 3D Shape Benchmark* and its corresponding BIC Score curve

show that an ”optimal” number of views exists where the BIC score is maximized. For the airplane model in figure 4(a), the number of optimal views is 29. For the car model in figure 5(a), only 17 views are needed as this 3D model is less complex then the first one. This is coming from the fact that, more the 3D model is geometrically complex, the more its 2D views are different. This leads to a higher number of views to best represent it.

3 Probabilistic approach for 3D indexing

As mentioned before, all the views of the same 3D models do not have the same importance. There are views that represent better the 3D model then others. On the other hand, simple objects (e.g. cube, sphere) can be at the root of more complex objects, they are more common. In this section, we present a probabilistic approach that takes into account that views do not have the same importance, and that simple objects have bigger probability to appear then more complex one.

Each model of the collection $D_b = \{M_1, M_2, \ldots, M_N\}$ is represented by a set of characteristic views $V_c = \{V_1, V_2, \ldots, V_{\hat{v}}\}$, with $\hat{v}$ the number of characteristic views. To each characteristic view corresponds a set of represented views called V_r.

Considering a 3D request model Q, we wish to find the model $M_i \in D_b$ which is the closest to the request model Q. This model is the one that has the highest probability $P(M_i|Q)$.

Knowing that each model is represented by its characteristic views, $P(M_i|Q)$ can be written:

$$P(M_i|Q) = \sum_{k=1}^{K} P(M_i|V_Q^k)P(V_Q^k|Q) \tag{4}$$

With K the number of characteristic views of the model Q. Let H^k be the set of all the possible hypotheses of correspondence between the request view V_Q^k and a model M_i, $H^k = \{h_1^k \vee h_2^k \vee \ldots \vee h_N^k\}$. A hypothesis h_p^k means that the view p of the model correspond to the view request V_Q^k. The sign $\vee$ represents *logic or operator*. Let us note that if an hypothesis h_p^k is true, all the other hypotheses are false. $P(M_i|Vc_Q^k)$ can be expressed by $P(M_i|H^k)$.

We have :

$$P(M_i|H^k) = \sum_{j=1}^{\hat{v}} P(M_i, V_{M_i}^j|H^k) \tag{5}$$

$$P(M_i|H^k) = \sum_{j=1}^{\hat{v}} \sum_{p=1}^{\hat{v}} P(M_i, V_{M_i}^j|h_p^k) \tag{6}$$

Knowing that we cannot observe in the same scene 2 characteristic views of the same 3D model, we have :

$$P(M_i, V_{M_i}^j|h_p^k) = 0 \text{ for every } j \neq p$$

Then :

$$P(M_i|H^k) = \sum_{j=1}^{\hat{v}} P(M_i, V_{M_i}^j|h_j^k) \tag{7}$$

The sum $\sum_{j=1}^{\hat{v}} P(M_i, V_{M_i}^j | h_j^k)$ can be reduced to the only true hypothesis $P(M_i, V_{M_i}^j | h_j^k)$. In fact, a characteristic view from the request model Q can match only one characteristic view from the model M_i . We choose the characteristic view with the maximum probability:

$$P(M_i|H^k) = Max_j P(M_i, V_{M_i}^j | h_j^k) \tag{8}$$

$$P(M_i|Q) = \sum_{k=1}^{K} Max_j (P(M_i, V_{M_i}^j | h_j^k)) P(V_Q^k|Q) \tag{9}$$

The closest model is the one that contains the view having the highest probability.

We have :

$$P(M_i, V_{M_i}^j, h_j^k) = P(h_j^k, V_{M_i}^j | M_i) P(M_i)$$

and,

$$P(M_i, V_{M_i}^j, h_j^k) = P(V_{M_i}^j, M_i | h_j^k) P(h_j^k)$$

Using the Bayes theorem, we obtain :

$$P(M_i, V_{M_i}^j | h_j^k) = \frac{P(h_j^k, V_{M_i}^j | M_i) P(M_i)}{P(h_j^k)} \tag{10}$$

On the other hand,

$$P(h_j^k, V_{M_i}^j | M_i) = P(h_j^k | V_{M_i}^j, M_i) P(V_{M_i}^j | M_i) \tag{11}$$

and,

$$P(h_j^k) = \sum_{i=1}^{N} \sum_{j=1}^{\hat{v}} P(h_j^k | V_{M_i}^j, M_i) P(V_{M_i}^j | M_i) P(M_i)) \tag{12}$$

By integrating this remark we obtain :

$$P(M_i, V_{M_i}^j | h_j^k) = \frac{P(h_j^k | V_{M_i}^j, M_i) P(V_{M_i}^j | M_i) P(M_i)}{\sum_{i=1}^{N} \sum_{j=1}^{\hat{v}} P(h_j^k | V_{M_i}^j, M_i) P(V_{M_i}^j | M_i) P(M_i)} \tag{13}$$

Finally:

$$P(M_i, V_{M_i}^j | h_j^k) = \sum_{k=1}^{K} Max_j (\frac{P(h_j^k | V_{M_i}^j, M_i) P(V_{M_i}^j | M_i) P(M_i)}{\sum_{i=1}^{N} \sum_{j=1}^{\hat{v}} P(h_j^k | V_{M_i}^j, M_i) P(V_{M_i}^j | M_i) P(M_i)}) P(V_Q^k|Q) \tag{14}$$

With $P(M_i)$ the probability to observe the model M_i.

$$P(M_i) = \frac{\alpha e^{(-\alpha.N(V_{M_i})/N(V))}}{\sum_{i=1}^{N} \alpha e^{(-\alpha.N(V_{M_i})/N(V))}} \tag{15}$$

Where $N(V_{M_i})$ is the number of characteristic views of the model M_i, and $N(V)$ is the total number of characteristic views for the set of the models of the collection D_b. α is a parameter to hold the effect of the probability $P(M_i)$. The algorithm conception makes that, the greater the number of characteristic views of an object, the more it is complex. Indeed, simple object (e.g. a cube) can be at the root of more complex objects.

On the other hand:

$$P(V_{M_i}^j|M_i) = \frac{1 - \beta e^{(-\beta . N(Vr_{M_i}^j)/N(Vr_{M_i}))}}{\sum_{j=1}^{\hat{v}}(1 - \beta e^{(-\beta . N(Vr_{M_i}^j)/N(Vr_{M_i}))})} \tag{16}$$

Where $N(Vr_{M_i}^j)$ is the number of views represented by the characteristic view j of the model M, and $N(Vr_{M_i})$ is the total number of views represented by the model M_i. The β coefficient is introduced to reduce the effect of the view probability. We use the values $\alpha = \beta = 1/100$ which give the best results during our experiments. The greater is the number of represented views $N(Vr_{M_i}^j)$, the more the characteristic view $V_{M_i}^j$ is important and the best it represents the three-dimensional model.

The value $P(h_j^k|V_{M_i}^j, M_i)$ is the probability that, knowing that we observe the characteristic view j of the model M_i, this view corresponds to the view k of the request model Q:

$$P(h_j^k|V_{M_i}^j, M_i) = \frac{e^{-D(V_Q^k, V_{M_i}^j)}}{\sum_{j=1}^{\hat{v}} e^{-D(V_Q^k, V_{M_i}^j)}} \tag{17}$$

With $D(V_Q^k, V_{M_i}^j)$ the Euclidean distance between the 2D Zernike descriptors of the view k of the request model Q and $V_{M_i}^j$ the characteristic view j of the three-dimensional model M_i.

To summarize, in this section we presented our Bayesian retrieval framework, which takes into account the number of characteristic views of the model and the importance (amount of information) of its views.

4 Experimental setup

In this section, we present the experimental results, the databases and the evaluation criteria. The algorithms we described in the previous sections have been implemented using C++ and the TGS OpenInventor libraries. The system consists in an off-line characteristic views extraction and an on-line retrieval process.

In our method AVC, every model was normalized for size by isotropically rescalling it so that the average euclidean distance from points on its surface to the center of mass is 0.5. Then, all the models was normalized for translation by moving its center of mass to the origin.

In the off-line process, the characteristic views selection takes about 18 seconds per model on PC with a Pentium IV 2.4 GHZ CPU. In the on-line process, the comparison takes less then 1 second for 1814 3D models. To evaluate our method, we used two different 3D model databases. The Princeton Shape Benchmark (PSB), a standard shape benchmark widely used in shape retrieval community. We also present our result on a more specific database, 5000 3D CAD models provided by the car manufacturer Renault within the framework of SEMANTIC-3D project[24].

During our experiments, and to show the contribution of the probabilistic approach to retieval performance, we compare the probabilistic correspondence between two 3D models to the use of a simple distance between the 3D models. We can define a distance between two 3D models by:

$$Dist(M_i, Q) = \sum_{j=1}^{\hat{v}} Min_k \ D(V_{M_i}^j, V_Q^k)$$

With $D(V_Q^k, V_{M_i}^j)$ the Euclidean distance between the 2D Zernike descriptors of the view k of the request model Q and $V_{M_i}^j$ the characteristic view j of the three-dimensional model M_i.

4.1 Data set

The next two sections present the 3D models databases we used in our experiments.

4.1.1 Princeton Shape Benchmark

Princeton Shape Benchmark (PSB) appeared in 2004 is one of the most exhaustive benchmarks for 3D shape retrieval. It contains a database of 1814 classified 3D models collected from 293 different Web domains. There are many classifications given to the objects in the database. During our experiments we used the finest granularity classification, composed of 161 classes. Most classes contain objects with a particular function (e.g cars). Yet, there are also cases where objects with the same function are partitioned in different classes based on their shapes (e.g round tables versus rectangular tables) [25]. We compare our method to some state of the art shape matching algorithms:

- **D2 Shape Distribution:** a histogram of distances between pairs of points on the surface [26].
- **Extended Gaussian Image (EGI):** A spherical function giving the distribution of surface normals [27].
- **Complex Extended Gaussian Image (CEGI):** a complex-valued spherical function giving the distribution of normals and associated normal distances of points on the surface [28].
- **Q-Gram Statistics Descriptor (2-GR):** a vector in 256-dimensional space, whose i-th co-ordinate is the number of Q_i 2-grams in the voxel representation of 3D Shape [29].
- **Shape Histogram (SHELLS):** a histogram of distances from the center of mass to points on the surface [30].
- **Shape Histogram (SECTORS):** a spherical function giving the distribution of model area as a function of spherical angles [30].
- **Shape Histogram (SECSHELL):** a collection of spherical functions that give the distribution of model area as a function of radius and spherical angles [30].
- **Voxel:** a binary rasterization of the model boundary into a voxel grid which is represented by the 32 spherical descriptors representing the intersection of the voxel grid with the concentric spherical shells.
- **Spherical Extent Function (EXT):** a spherical function giving the maximal distance from center of mass as a function angle [31].
- **Radialized Spherical Extent Function (REXT):** a collection of spherical functions giving the maximal distance from center of mass as a function of spherical angle and radius [32].
- **Gaussian Euclidean Distance Transform (GEDT):** a 3D function whose value at each point is given by composition of a Gaussian with the Euclidean Distance Transform of the surface [1].

- **Spherical Harmonic Descriptor (SHD):** a rotation invariant representation of the GEDT obtained by computing the restriction of the function to co-centric spheres and storing the norm of each (harmonic) frequency [1].

- **Light Field Descriptor (LFD):** a representation of a model as a collection of images rendered from uniformly sampled positions on a view sphere. The distance between two descriptors is defined as the minimum $L1$ difference, taken over all rotations and all pairings of vertices on two dodecahedra [12].

Every model was normalized for size by isotropically rescaling it so that the average distance from points on its surface to the center of mass is 0.5. Then for all descriptors except D2 and EGI, the model was normalized for translation by moving its center of mass to the origin. Next, for all descriptors except D2, SHELLS, SHD and LFD, the model was normalized for rotation by aligning its principal axes to the x-, y-, and z-axes. The ambiguity between positive and negative axes was resolved by choosing the direction of the axes so that the area of model on the positive side of the x-, y-, and z-axes was greater than the area on the negative side. Every spherical descriptor(EGI, CEGI, Sectors, etc.), was computed on a 64x64 spherical grid and represented by its harmonic coefficients up to order 16. Histograms of distances (D2 and Shells) was stored with 64 bins representing distances in the range [0,2]. All descriptors, except LFD, were scaled to have L_2-norm equal to 1. The LFD comprises 100 images encoded with 35, 8-bit, coefficients to describe Zernike moments and 10, 8-bit, coefficients to represent Fourier descriptors. For more details about PSB-2004 refer to [25].

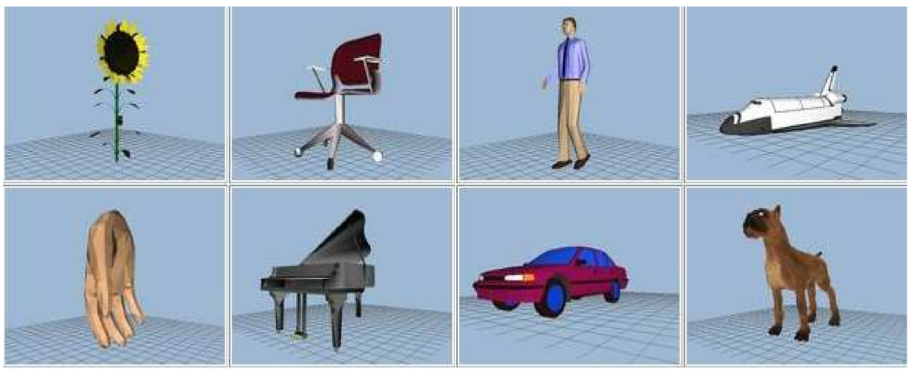


Figure 6: 3D models from Princeton Shape Benchmark[25].

4.1.2 SEMANTIC-3D Database

The SEMANTIC-3D project[24], supported by the French Research Ministry and the RNRT aims the exploration of techniques and tools preliminary to the realization of new services for the exploitation of 3D contents through the Web. New techniques of compression, indexing and watermarking of 3D data are developed and implemented in a industrial prototype application: a communication and information system(consultation, remote support) between the authors (originators of machine elements), users (car technician) and a central server of 3D data. The compressed 3D data exchanges adapt to the rate of transmission and the capacity of the terminals used.

These exchanges are ensured with a format of data standardized with functionalities of visualization and 3D animation. The exchanges are achieved on existing networks (internal network of the authors, wireless networks for the technicians).

In SEMANTIC-3D project, our research group focuses on the indexing and retrieval of 3D data.

Within the framework of SEMANTIC-3D project, a database of 3D CAD models was provided by the car manufacturer Renault. The 3D models database contains 5000 of quite irregular polygonal meshes representing CAD parts. To measure the performance, we classified the 758 models of this collection into 75 classes based on the judgment of two adult persons (figure 4.1.2), the 4242 models were put in a class called "others". The smallest class contain four 3D models. The largest class contains sixty four 3D models. Figure 4.1.2 shows some 3D models from the SEMANTIC-3D Database.

4.2 Evaluation criterion

To objectively evaluate our method we use a statistical tool provided by Princeton shape benchmark (PSB) to compare 3D retrieval methods. Given a classification and a distance matrix computed with any shape matching algorithm, a suite of PSB Benchmark tools produces statistics and visualizations that facilitate the evaluation of the match results [25].

The statistics that can be calculated using the PSB tools are:

- **Nearest neighbor:** the percentage of the closest matches that belong to the same class as the query. This statistic provides an indication of how well a nearest neighbor classifier

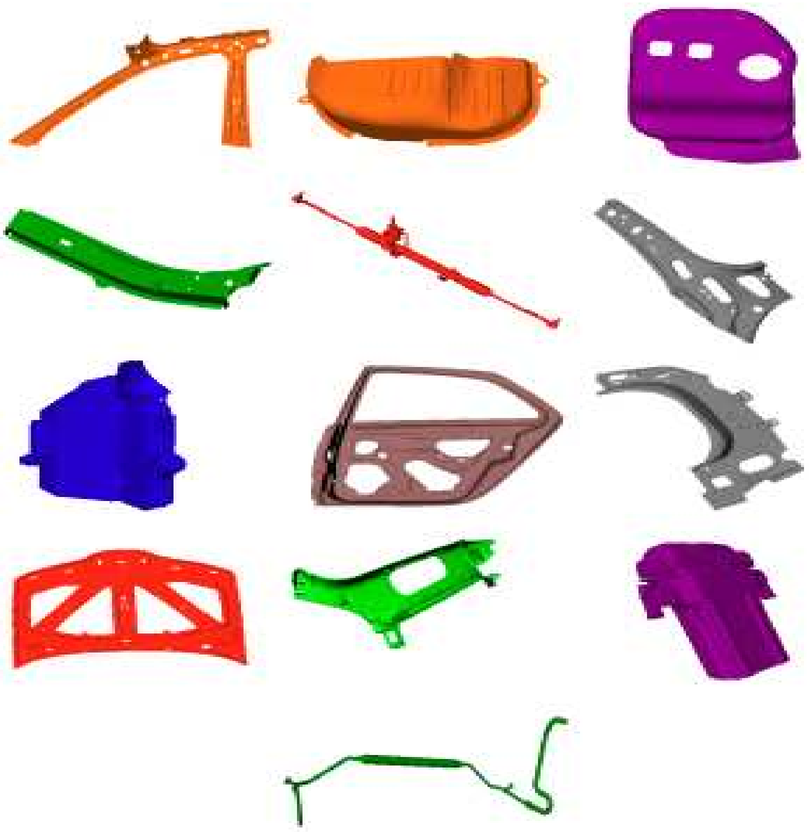


Figure 7: 3D models from SEMANTIC-3D database.

would perform. Obviously, an ideal score is 100%, and higher scores represent better results.

- **First-tier and Second-tier:** the percentage of models in the query's class that appear within the top K matches, where K depends on the size of the queries class. Specifically, for a class with $|C|$ members, $K = |C| - 1$ for the first tier, and $K = 2 \times (|C| - 1)$ for the second tier. The first tier statistic indicates the recall for the smallest K that could possibly include 100% of the models in the query class, while the second tier is a little less stringent (i.e., K is twice as big). These statistics are similar to the Bulls Eye Percentage Score $K = 2 \times |C|$, which has been adopted by the MPEG-7 visual SDs. In all cases, an ideal matching result gives a score of 100%, and higher values indicate better matches.

- **E-Measure:** a composite measure of the precision and recall for a fixed number of retrieved results. The intuition is that a user of a search engine is more interested in the first page of query results than in later pages. So, this measure considers only the first 32 retrieved models for every query and calculates the precision and recall over those results. The E-Measure is defined as:

$$E = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

The E-measure is equivalent to subtracting van Rijsbergen's definition of the E-measure from 1. The maximum score is 1.0, and higher values indicate better results

- **Discounted Cumulative Gain (DCG):** a statistic that weight correct results near the front of the list more than correct results later in the ranked list under the assumption that a user is less likely to consider elements near the end of the list. Specifically, the ranked list R is converted to a list G, where element G_i has a value 1 if element R_i is in the correct class and 0 otherwise. Discounted cumulative gain is then defined as follows :

$$DCG_1 = G_1; DCG_i = DCG_{i-1} + \frac{G_i}{lg_2(i)}, if i > 1$$

The result is then divided by the maximum possible DCG (i.e, that would be achieved if the first C elements were in the correct class, where C is the size of the class) to give the final score :

$$DCG = \frac{DCG_k}{1 + \sum_{j=2}^{|C|} \frac{1}{lg_2(j)}}$$

where k is the number of models in the database. The entire query result list is incorporated in an intuitive manner by the discounted cumulative gain [33].

- **Recall and Precision Curves:** Recall and Precision Curves are well known in the literature of content-based search and retrieval. The recall and precision are defined as follow:

$$\text{Recall} = \text{N/Q}, \text{ Precision} = \text{N/A}$$

With N the number of relevant models retrieved in the top A retrievals. Q is the number of relevant models in the collection, that are, the class number of models to which the query belongs to.

5 Experiments and results

5.1 Princeton Shape Benchmark

In our experiment, we computed the distances between all pairs of models in the *Princeton 3D Shape Benchmark* and analyze them with the *Princeton 3D Shape Benchmark* evaluation tools to quantify the matching performance with respect to the base classification.

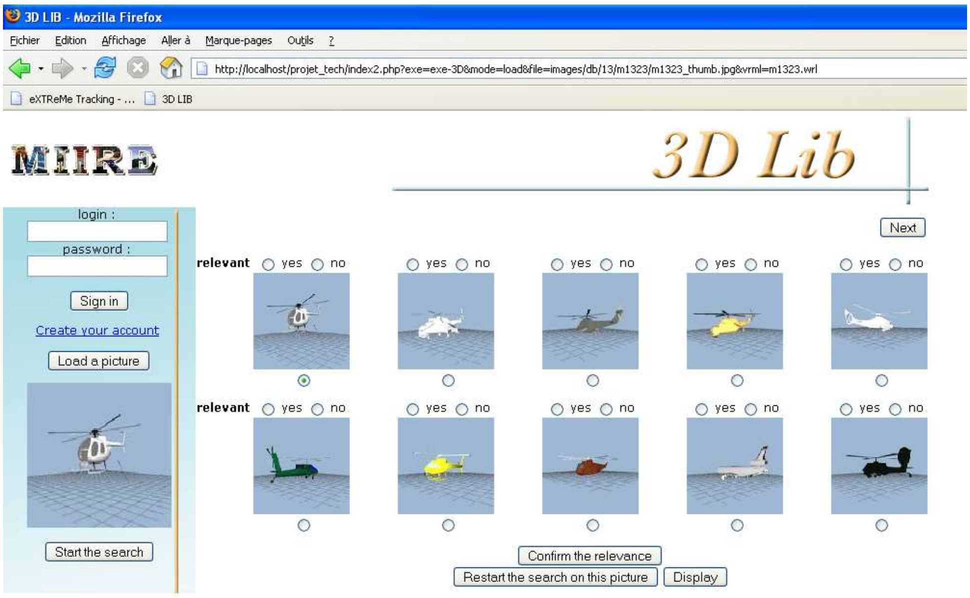


Figure 8: Screenshot of the 3D models retrieval system.

Figure 8 shows a request using our 3D retrieval system. On the left side, the request 3D model is presented. The right side shows the 3D models which have the highest probabilities of matching the 3D request model.

As mentioned before, we use several different performance measures to objectively evaluate our method: the First Tier (FT), Second Tier (ST), Nearest Neighbor (NN), E-Measure, Discounted Cumulative Gain (DCG) and Normalized Discounted Cumulative Gain (N-DCG) match percentages, as well as the recall-precision plot [34].

Table 1 shows micro averages storage requirement (for our method, we used 23 views that is the average number of characteristic views for all the database models) and retrieval statistics for each algorithm. Storage size is given in bytes. We found that micro and macro-average results gave consistent results, and we decided to present micro-averaged statistics.

Methods	Discrimination						
	Storage size	NN	FT	ST	E-Measure	DCG	N-DCG
LFD	4,700	65.7%	38.0%	48.7%	28.0%	64.3%	21.3%
AVC(probability)	**1,113**	**60.6%**	**33.2%**	**44.3%**	**25.5%**	**60.2%**	**13.48%**
REXT	17,416	60.2%	32.7%	43.2%	25.4%	60.1%	13.3%
GEDT	32,776	60.3%	31.3%	40.7%	23.7%	58.4%	10.2%
AVC(simple distance)	**1,113**	**58.2%**	**31.1%**	**42.7%**	**25.1%**	**59.9%**	**11,8%**
2-GR	512	55.5%	28.7%	39.1%	23.0%	56.3%	—%
EXT	552	54.9%	28.6%	37.9%	21.9%	56.2%	6.0%
SECSHEL	32,776	54.6%	26.7%	35.0%	20.9%	54.5%	2.8%
VOXEL	32,776	54.0%	26.7%	35.3%	20.7%	54.3%	2.4%
SECTORS	552	50.4%	24.9%	33.4%	19.8%	52.9%	-0.3%
CEGI	2,056	42.0%	21.1%	28.7%	17.0%	47.9%	-9.6%
EGI	1,032	37.7%	19.7%	27.7%	16.5%	47.2%	-10.9%
D2	136	31.1%	15.8%	23.5%	13.9%	43.4%	-18.2%
SHELLS	136	22.7%	11.1%	17.3%	10.2%	38.6%	-27.3%

Table 1: Retrieval performances for Princeton Shape Benchmark

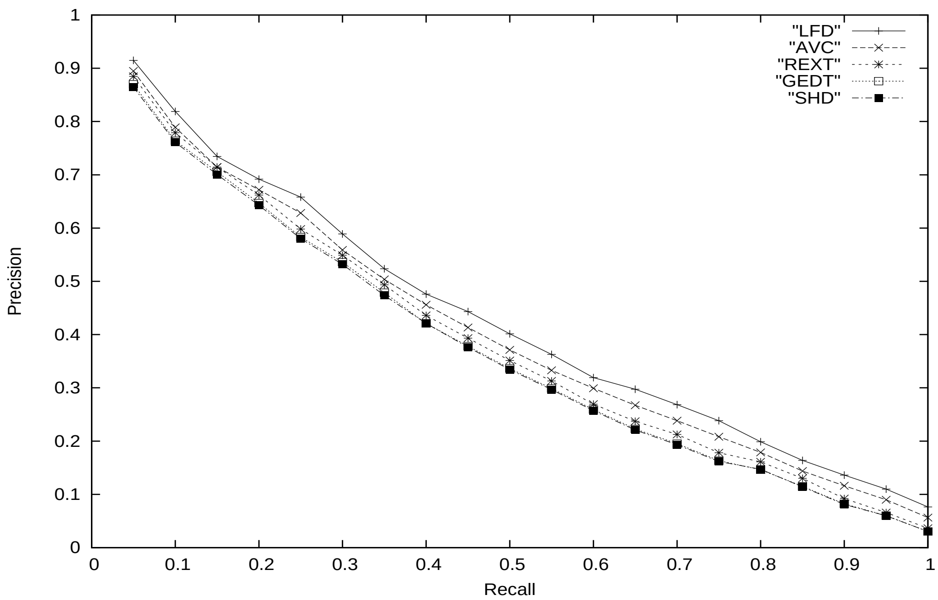


Figure 9: Recall Precision on *Princeton 3D Shape Benchmark* database.

Figure 9 shows the recall precision plots for our method AVC and some other shape descriptors. We presented the curves of the most relevant algorithm only, to keep the figure clear.

We find that the shape descriptors based on 2D views (LFD and AVC) provides the best retrieval precision in this experiment. We might expect shape descriptors that capture 3D geometric relationships would be more discriminating than the ones based solely on 2D projections, the opposite is true.

We can notice that our method provides more accurate results with the use of Bayesian probabilistic indexing. The experiment shows that AVC gives better performances then 3D harmonics, Radialized Spherical Extent Function and Gaussian Euclidean Distance Transform on the *Princeton 3D Shape Benchmark* database. Light Field Descriptor gives better results than our method but uses 100 views, does not adapt the number of views to the geometrical complexity and uses two descriptors for each view (Zernike moments and Fourier descriptor), which make it slower and more memory consuming descriptor compared to the method we presented.

Overall, we can conclude that AVC gives a good compromise between quality (relevance) / cost (memory and on-line comparison time) between the shape descriptors we compared to using the *Princeton 3D Shape Benchmark*.

5.2 SEMANTIC-3D Database

The experiments are made on a database that contains 5000 3D models. To objectively evaluate the performance of our method on this database, a classification was made (ground truth). 758 models are classified on 75 classes. A "special" class called others contain all the 3D models that was not classified.

Figure 10: A query 3D Model.

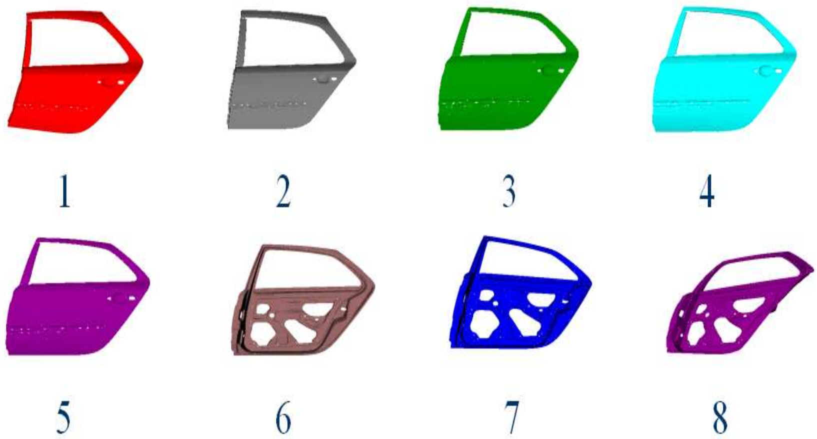


Figure 11: Results of the query presentend in figure 6

Figure 10 shows 3D model from SEMANTIC-3D database that is used as a query model to our system. Figure 11 shows the ranked results to the query in figure 10.

Methods	Discrimination				
	NN	FT	ST	E-Measure	DCG
AVC(probability)	**99.2%**	**88.3%**	**96.7%**	**52%**	**96.3%**
AVC(simple distance)	**98.1%**	**86.8%**	**95.1%**	**51.3%**	**93.9%**

Table 2: Retrieval performances for SEMANTIC-3D database

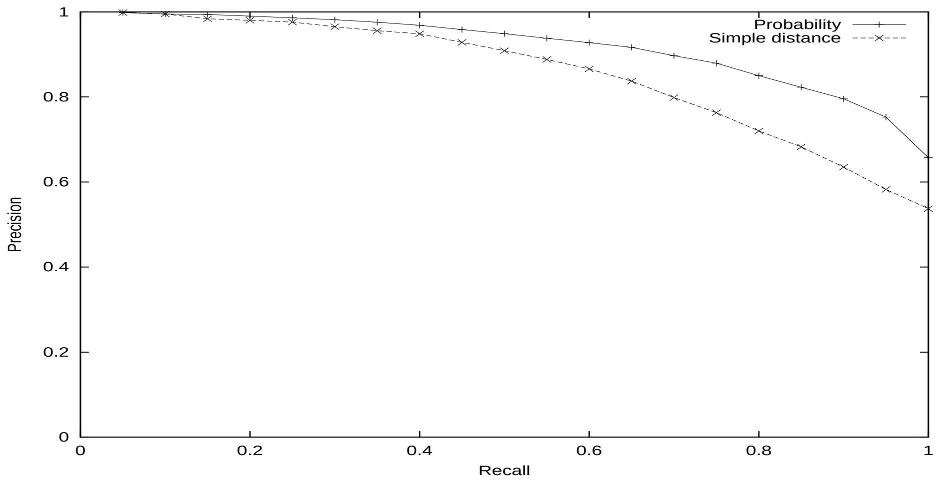


Figure 12: Recall Precision for Semantic-3D database.

Figure 12 shows the recall precision plots for our method AVC on the SEMANTIC-3D database. We can explain the good results of our method by the fact that the variance intra-class are very small. As the database contains real professional 3D CAD Models, the 3D models from the same class represent real mechanical parts used in a car. The 3D models from the same class represent different versions of the same models with small changes.

In the other hand, 3D CAD models in the database contains holes so that they can be fixed to other mechanical parts. The positions and the dimension of the holes can differentiate between two different models from the same class. As we represent each view of the 3D model by Zernike moments, the holes and the global shape are well taken into account. We can also notice the result enhancement when we use the probabilistic approach for retrieval.

5.3 Robustness evaluation of the method

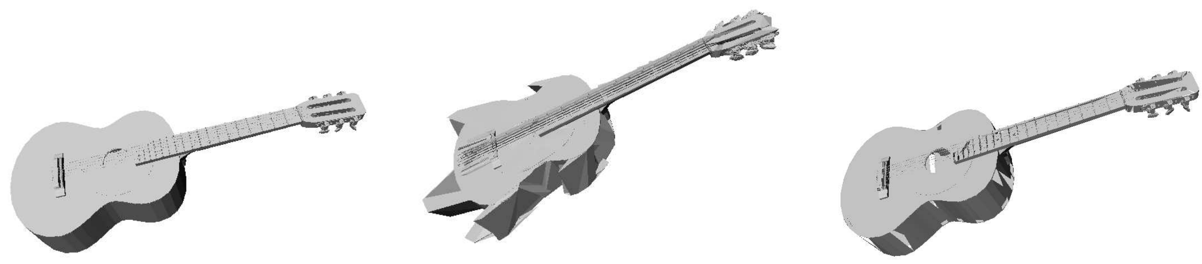
In order to assess the robustness of our method, we apply the following transformations for all classified 3D models. Each transformed 3D model is then used to query the test database.

The average recall and precision of all the 1814 classified models from PSB are used for the evaluation (figure 14). The robustness is evaluated by the following transformation:

1. noise: each vertex of 3D model is applied three random number to x-, y- and z-axis translation ($\pm 15\%$ times of the length of the model's bounding box). Figure 13(b) shows a typical example of the noise effect.

2. decimation: for each 3D model, randomly select 20% polygons to be deleted. Figure 13(c) shows a typical example of the decimation effect.

Experimental results of the robustness evaluation is shown in Figure 14. These experimental results prove that our approach is robust against noise and decimation.



(a) original 3D model (b) noise (c) decimation

Figure 13: Robustness evaluation of noise and decimation from a 3D model.

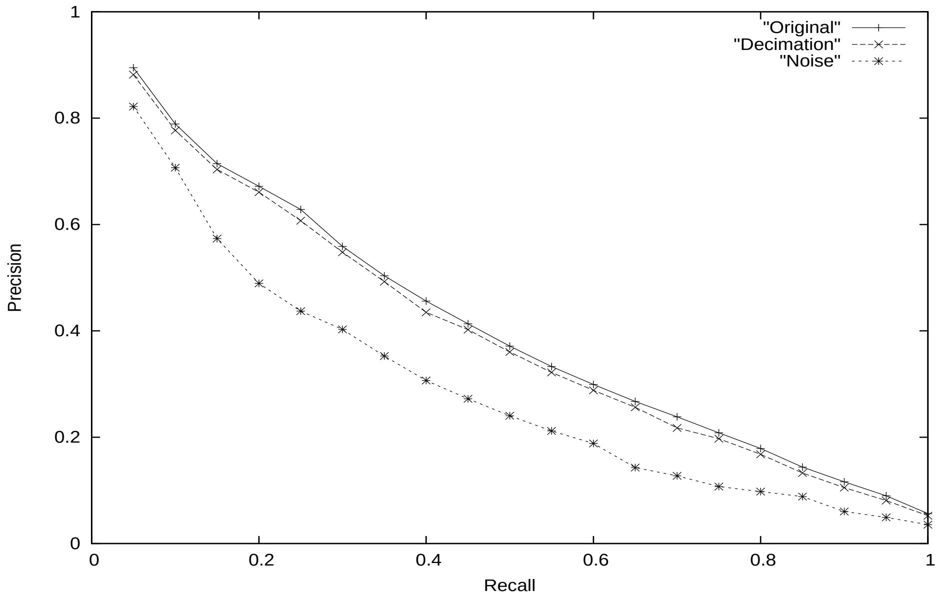


Figure 14: Recall Precision on Princeton 3D Shape Benchmark database with deformations effect.

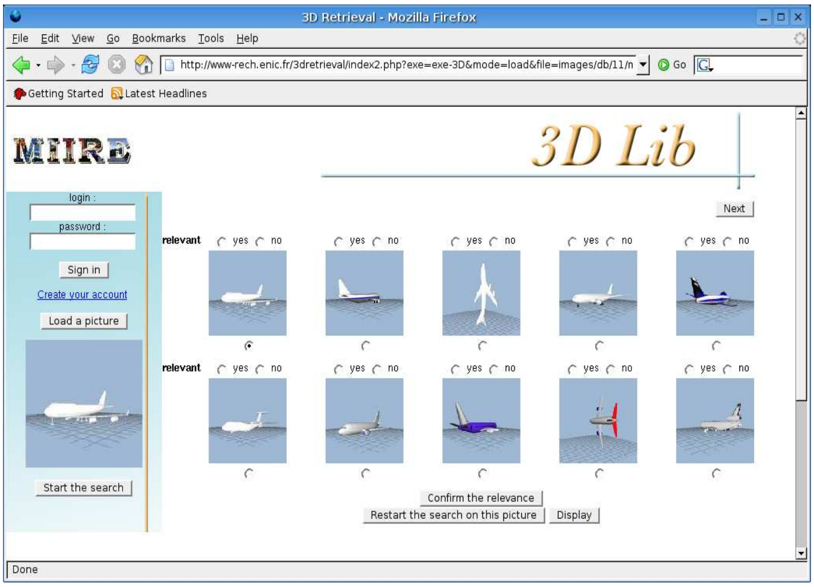
5.4 On-line Search Engine

The SEMANTIC-3D project focuses on the development of tools and methods required to implement new operational services for retrieving 3D content through the Web and communicating objects. Information and communication system must be available for remote access and assistance, interconnecting originators (mechanical part designers), nomadic users (automotive industry technicians) and a central 3D data server.

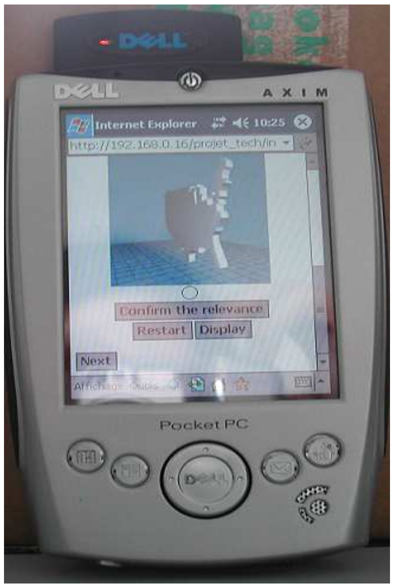
To experiment our algorithms and to asset the results presented in the previous sections, we developed an on-line 3D search engine. Our search engine can be reached from any device having compatible web browser (PC, PDA, SmartPhone, etc.) [35].

Figure 15(a) shows the interface of our system using a PC web browser. Figure 15(b) shows the interface of our system on PDA (Pocket PC under Windows Pocket 2003).

Depending on the web access device he/she is using, the user faces two different kind of web interfaces : a rich web interface for full-featured web browsers (figure 15(a)), and a simpler interface for PDA web browsers (figure 15(b)). In both cases, the results returned by the 3D search engine are the same. The only difference lies in the design of the results presentation.



(a) PC browser interface



(b) PDA browser interface

Figure 15: The PC and PDA browser interface of our on-line search engine.

At last, the users should notice that, due to some copyright protection, the 3D CAD models from the SEMANTIC-3D project are not available for public on-line use. The 3D database available for tests of our 3D search engine is the **Princeton Shape Benchmark** Database [25].

6 Discussion and Conclusion

In this paper, we propose a 3D model retrieval system based on characteristic views similarity called AVC (Adaptive Views Clustering). Starting from the fact that the more the 3D model is geometrically complex, the more its 2D views are different, we propose a characteristic views selection algorithm that corresponds the number of views to its geometrical complexity. Starting from 320 initial views, our algorithm select the "optimal" characteristic views set that best represent the 3D model. The number of characteristic views varies from 1 to 40. We also propose a new probabilistic retrieval approach that takes into account that not all the views of 3D models have the same importance, and also the fact that geometrically simple models have more probability to be relevant than more complex ones. Based on some standard measures, we compared our method to twelve state of the art methods on *Princeton 3D Shape Benchmark* database. Our method gives the second best results. The AVC method we proposed gives a good quality/cost compromise compared to other well-known methods. The good results of our method on a large 3D CAD models database (5000 models) supplied by *Renault*, show that our method can also be suitable for 3D CAD models retrieval. Our method is robust against noise and model degeneracy. It can be suitable against topologically ill-defined 3D models. A practical 3D models retrieval system based on our approach is available on the web for on-line tests [35].

7 Acknowledgments

This work is supported by the French Research Ministry and the RNRT (Réseau National de Recherche en Télécommunications) within the framework of the SEMANTIC-3D National Project (http://www.semantic-3d.net). We would like to thank Professor Dan Pelleg from Carnegie Mellon University for the help he provided on X-means.

References

[1] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3d shape descriptors," in *Symposium on Geometric Processing*, 2003.

[2] J. P. Vandeborre, V. Couillet, and M. Daoudi, "A practical approach for 3D model indexing by combining local and global invariants," in *1st IEEE International Symposium on 3D Data Processing, Visualization and Transmission*, June 2002, pp. 644–647.

[3] G. Antini, S. Berretti, A. D. Bimbo, and P. Pala, "Retrieval of 3d objects using curvature correlograms," in *ICME*, july 2005.

[4] J. Assfalg, A. D. Bimbo, and P. Pala, "Spin images for retrieval of 3d objects by local and global similarity," in *ICPR*, vol. 3, August 2004, pp. 906–909.

[5] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton based shape matching and retrieval," in *IEEE proceedings of the Shape Modeling International*, 2003.

[6] T. Funkhouser, P. Min, M. Kazhdan, A. Haderman, D. Dobkin, and D. Jacobs, "A search engine for 3D models," *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 83–105, 2003.

[7] T. F. Ansary, J. P. Vandeborre, and M. Daoudi, "Bayesian approach for 3D models retrieval based on characteristic views," in *17th IEEE International Conference On Pattern Recognition*, August 2004.

[8] T. F. Ansary, J. P. Vandeborre, S. Mahmoudi, and M. Daoudi, "A bayesian framework for 3D models retrieval based on characteristic views," in *2nd IEEE International Symposium on 3D Data Processing, Visualization and Transmission*, September 2004.

[9] C. Cyr and B.Kimia, "3D object recognition using shape similarity-based aspect graph," in *ICCV01*, 2001, pp. 254–261.

[10] R. Schiffenbauer, "A survey of aspect graphs," CIS, Tech. Rep. TR-CIS-2001-01, 2001.

[11] D. Macrini, A. Shokoufandeh, S. Dickenson, K. Siddiqi, and S.Zucker, "View based 3-D object recognition using shock graphs," in *ICPR*, vol. 3, 2002, pp. 24–28.

[12] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," in *Eurographics*, vol. 22, 2003, pp. 223–232.

[13] J. Tangelder and R. Veltkamp, "A survey of content based 3D shape retrieval methods," in *Shape Modeling International*, 2004, pp. 145–156.

[14] A. Khotanzad and Y. Hong, "Invariant image recognition by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489 – 497, May 90.

[15] W. Kim and Y. Kim, "A region-based shape descriptor using Zernike moments," *Signal Processing: Image Communication*, vol. 16, pp. 95–100, 2000.

[16] R. Duda and P. Hart, "Pattern classification and scene analysis," *John Wiley and Sons*, 1973.

[17] D. Pelleg and A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *International Conference on Machine Learning*, 2000, pp. 727–734.

[18] H. Akaike, "Information theory and extention of the maximum likehood principle," in *International symposium on information theory*, 1973, pp. 267–281.

[19] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[20] J. Berger and L. Pericchi, "The intrinsic bayes factor for model selection and prediction," *The Journal of the American Statistical Association*, pp. 109–122, 1994.

[21] K. Burnham and D. Anderson, *Model selection and multi-model inference.* Springer-Verlag, 2002.

[22] H. Bozdogan, "Model selection and akaike's information criteria (aic): the genral theory and its analytical extentions," *Psychometrika 52*, pp. 354–370, 1987.

[23] E. Lebarbier and T. Mary-Huard, "Le critère bic: fondements théoriques et interprétaiton," INRIA, Tech. Rep. RR5315, 2004.

[24] "Semantic-3d project web site," in *http://www.semantic-3d.net.*

[25] "Princeton shape benchmark (2004)," in *http://shape.cs.princeton.edu/benchmark.*

[26] R. Osada, T. Funkhouser, B. Chazells, and D. Dobkin, "Matching 3D models with shape distributions," in *Shape Modeling International*, May 2001.

[27] B. Horn, "Extended gaussian images," *Proceedings of the IEEE 72*, vol. 12, pp. 1671–1686, December 1984.

[28] S. Kang and K. Ikeuchi, "Determining 3d object pose usiing the complex extended gaussian image," in *CVPR*, June 1991, pp. 580–585.

[29] E. Ivanko and D. Pervalov, "Q-gram statistics descriptor in 3d shape classification," in *Internation Conference on Advances in Pattern Recognition*, August 2005, pp. 360–367.

[30] M. Ankerst, G. Kastenmuller, H.-P. Kriegel, and T. Seidl, "Nearest neighbor classification in 3d protein databases," in *Proc. ISBM*, 1999.

[31] D. Saupe and D. V. Vranic, "3d model retrieval with spherical harmonics and moments," in *In B. Radig and S. Florczyk, editors, DAGM 2001*, September 2001, pp. 392–397.

[32] D. Vranic, "An improvement of rotation invariant 3d shape descriptor based on functions on concentric spheres," in *ICIP*, 2003, pp. 757–760.

[33] G. Leifman, S. Katz, A. Tal, and R. mei, "Signatures of 3d models for retrieval," in *The 4th Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics*, August 2003, pp. 159–163.

[34] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The princeton shape benchmark," in *Shape Modeling International*, 2004.

[35] "Miire - 3d search engine," in *http://www-rech.enic.fr/3dretrieval*.